The Consultative Committee for Space Data Systems

Draft Recommendation for
Space Data System Standards

CCSDS FILE DELIVERY PROTOCOL (CFDP)

Draft Recommended Standard

CCSDS 727.0-~~B~~P-3.1

~~BLUE BOOK~~Pink Sheets
~~June 2005~~October 2006

5 PDU FORMATS

5.1 GENERAL

5.1.1 PDUs shall consist of a fixed-format PDU header followed by a fixed- or variable-format PDU data field.

5.1.2 The PDU header shall consist of the fields shown in table 5-1. Optional fields are present only when so indicated by the corresponding flag bit.

5.1.3 The PDU header fields shall be transmitted in the order of presentation in table 5-1.

NOTE – Every PDU is associated with a unique transaction by the source Entity ID together with the Transaction sequence number.

Table 5-1: Fixed PDU Header Fields

Field	Length (bits)	Values	Comment
Version	3	'000'	For the first version.
PDU type	1	'0' — File Directive '1' — File Data	
Direction	1	'0' — toward file receiver '1' — toward file sender	Used to perform PDU forwarding.
Transmission Mode	1	'0' — acknowledged '1' — unacknowledged	
CRC Flag	1	'0' — CRC not present '1' — CRC present	
Reserved for future use	1	set to '0'	
PDU Data field length	16		In octets.
Reserved for future use	1	set to '0'	
Length of entity IDs	3		Number of octets in entity ID less one; i.e., '0' means that entity ID is one octet. Applies to all entity IDs in the PDU header.
Reserved for future use	1	set to '0'	
Length of Transaction sequence number	3		Number of octets in sequence number less one; i.e., '0' means that sequence number is one octet.
Source entity ID	variable		Uniquely identifies the entity that originated the transaction. A binary integer.
Transaction sequence number	variable		Uniquely identifies the transaction, among all transactions originated by this entity. A binary integer.
Destination entity ID	variable		Uniquely identifies the entity that is the final destination of the transaction's metadata and file data. A binary integer.

5.1.4 Every entity ID in any CFDP PDU shall be represented as an unsigned binary integer of length as indicated by an entity ID length field elsewhere in the PDU. Every transaction sequence number in any CFDP PDU shall be represented as an unsigned binary integer of length as indicated by a transaction sequence number length elsewhere in the PDU.

5.1.5 The sequence number shall be sequentially allocated by the CFDP Entity responding to the **`Put.request`**.

5.1.6 Two PDU types are defined:

- File Directive;
- File Data.

5.1.7 The PDU Data field can contain data items in any one of four formats:

- fixed-format data;
- Length, Value (LV) format;
- Type, Length, Value (TLV) format;
- variable-length file data.

NOTES

1 Fixed-format data is precisely defined in this document and is optimized for efficiency of the protocol where the format of the data is not required to be variable. Fixed format data fields are fixed in their position in the PDU and in their length.

2 Each PDU field referred to as an offset is encoded as the number of octets in the file prior to the octet described by that field; e.g., the offset of the first octet of the file is encoded as all 'zeroes'.

3 Entity IDs may be of different lengths in different PDUs, as may transaction sequence numbers, but these differences in length have no semantic significance. Two entity IDs of different lengths are compared by, in effect, inserting leading 'zeroes' to pad the shorter entity ID to the same length as the longer entity ID and then comparing the resulting values. The same procedure applies to comparisons between transaction sequence numbers of different lengths.

5.1.8 The general format of LV objects shall be as shown in table 5-2.

Table 5-2: LV Object Format

Field	Size (bits)	Values	Comment
Length	8	0 to 255	Length of value.
Value	8 × Length		

5.2 FILE DIRECTIVE PDUs

5.2.1 GENERAL

5.2.1.1 The data field of File Directives shall consist of a Directive Code octet followed by a Directive Parameter field.

5.2.1.2 The File Directive Code shall indicate one of the ~~actions~~directives shown in table 5-4.

Table 5-4: File Directive Codes

Directive Code (hexadecimal)	~~Action~~Directive
00	Reserved
01	Reserved
02	Reserved
03	Reserved
04	EOF PDU
05	Finished PDU
06	ACK PDU
07	Metadata PDU
08	NAK PDU
09	Prompt PDU
0C	Keep Alive PDU
0D–FF	Reserved

NOTE – The Directive Parameter fields for the above actions are described, in order of transmission, in the following subsections.

5.2.1.3 In several cases, the Directive Parameter field of a File Directive includes a four-bit Condition Code. The Condition Code shall in each case indicate one of the conditions shown in table 5-5.

5.2.5 METADATA PDU

The contents of the Parameter field for a File Directive having a Code of Metadata PDU shall be as shown in table 5-9.

Table 5-9: Metadata PDU Contents

Parameter	Length (bits)	Values	Comments
Segmentation control	1	'0' — Record boundaries respected '1' — Record boundaries not respected	
Reserved for future use	7		Set to all 'zeroes'.
File size	32	Length of File	In octets. Set to all 'zeroes' for a file of unbounded size.
Source file name	LV	~~LV Length field indicates zero length and LV value field omitted when there is no associated file, e.g., messages used for Proxy operations.~~	When there is no associated file, e.g., messages used for Proxy operations, the LV Length field indicates zero length and the LV value field is omitted.
Destination file name	LV	~~LV Length field indicates zero length and LV value field omitted when there is no associated file, e.g., messages used for Proxy operations.~~	When there is no associated file, e.g., messages used for Proxy operations, the LV Length field indicates zero length and the LV value field is omitted.
Options	TLVs		Filestore requests. Messages to user. Fault Handler overrides. Flow Label.

Table 6-2: Originating Transaction ID Message

Field	Size (bits)	Values	Comments
Reserved for future use	1	set to '0'	
Length of entity ID	3		Number of octets in source entity ID less one; i.e., '0' means that source entity ID is one octet.
Reserved for future use	1	set to '0'	
Length of Transaction sequence number	3		Number of octets in sequence number less one; i.e., '0' means that sequence number is one octet.
Source entity ID	variable		Uniquely identifies the entity that originated the transaction. A binary integer.
Transaction sequence number	variable		Uniquely identifies the transaction, among all transactions originated by this entity. A binary integer.

6.2.3.1.2 The CFDP user that is the destination of the file delivery unit (the respondent) shall in turn use the contents of these messages to formulate a second **`Put.request`** primitive that accomplishes the desired remote file delivery.

6.2.3.1.3 The transaction ID assigned to the original request, as reported to the originating CFDP user by the corresponding **`Transaction.indication`** primitive, shall be cited in the transaction that constitutes the proxy's eventual report on the results of the requested proxy operation.

6.2.3.2 Proxy Put Request

The Proxy Put Request message is mandatory and shall be constructed as indicated in table 6-4.

Table 6-4: Proxy Put Request Message

Field	Size (bits)	Values	Comments
Destination entity ID	Variable	LV	The ID of the beneficiary's CFDP entity. A binary integer.
Source file name	Variable	LV	Length is zero if parameter is omitted.
Destination file name	Variable	LV	Length is zero if parameter is omitted.

6.2.3.3 Proxy Message to User

One or more Proxy Message to User messages may be optionally included and shall be constructed as indicated in table 6-5.

Table 6-5: Proxy Message to User Message

Field	Size (bits)	Values	Comments
Message text	Variable	LV	See 5.4.3.

6.4.3.2 Remote Status Report Request

The Remote Status Report Request message shall be constructed as indicated in table 6-17.

Table 6-17: Remote Status Report Request Message

Field	Size (bits)	Values	Comments
Reserved for future use	1	set to '0'	
Length of entity ID	3		Number of octets in source entity ID less one; i.e., '0' means that source entity ID is one octet.
Reserved for future use	1	set to '0'	
Length of Transaction sequence number	3		Number of octets in sequence number less one; i.e., '0' means that sequence number is one octet.
Source entity ID	variable		Uniquely identifies the entity that originated the transaction for which a status report is requested. A binary integer.
Transaction sequence number	variable		Uniquely identifies the transaction, among all transactions originated by this entity. A binary integer.
Report File Name	Variable	LV	The file name and path at the filestore local to the requesting CFDP user in which the responding CFDP user should put the status report.

6.4.4 RESPONDING TO A REMOTE STATUS REPORT REQUEST

6.4.4.1 General

6.4.4.1.1 Upon receipt of a Remote Status Report Request message, the remote CFDP user shall use the CFDP **`Put.request`** primitive to request delivery of a file delivery unit to the entity which sent the original Status Report Request message; the file data of this file delivery unit shall be a file that contains the status report for the indicated transaction, while the metadata of this file delivery unit shall contain one each of the Reserved CFDP Messages of message types Remote Status Report Response (defined in 6.4.4.2) and Originating Transaction ID.

6.4.4.1.2 If the remote CFDP user is unable to provide a status report file, then the responding file delivery unit shall contain no file and the report response code in the Remote Status Report Response message shall indicate that the remote status report request was unsuccessful.

6.4.4.1.3 If the remote CFDP user is able to provide a status report file, the nature of the status report shall be as described in 3.3.8.

6.4.4.2 Remote Status Report Response

The Remote Status Report Response message shall be constructed as indicated in table 6-18.

Table 6-18: Remote Status Report Response Message

Field	Size (bits)	Values	Comments
Transaction status	2	Same values as defined for transaction status in the ACK PDU (see table 5-8)	
Spare	5	All 'zeroes'	
Report response code	1	'0' = unsuccessful '1' = successful	
Spare	1	'0'	This field and the following five fields identify the transaction whose status is being reported; they are taken from the status report request.
Length of Entity ID	3		
Spare	1	'0'	
Length of Sequence Number	3		
Source Entity ID	Variable		A binary integer.
Transaction Sequence Number	Variable		A binary integer.

6.4.5 RETRIEVING A REMOTE STATUS REPORT

Upon receiving the Reserved CFDP Message with a Remote Status Report Response message type, the requesting CFDP user may retrieve the report from the file specified in the message.

6.5.3.2 Remote Suspend Request

The Remote Suspend Request message shall be constructed as indicated in table 6-20.

Table 6-20: Remote Suspend Request Message

Field	Size (bits)	Values	Comments
Reserved for future use	1	set to '0'	
Length of entity ID	3		Number of octets in source entity ID less one; i.e., '0' means that source entity ID is one octet.
Reserved for future use	1	set to '0'	
Length of Transaction sequence number	3		Number of octets in sequence number less one; i.e., '0' means that sequence number is one octet.
Source entity ID	variable		Uniquely identifies the entity that originated the transaction that is to be suspended. A binary integer.
Transaction sequence number	variable		Uniquely identifies the transaction, among all transactions originated by this entity. A binary integer.

6.5.4 RESPONDING TO A REMOTE SUSPEND REQUEST

6.5.4.1 General

6.5.4.1.1 Upon receipt of a Remote Suspend Request message, the remote CFDP user shall assure the suspension of the indicated transaction if possible; the CFDP **`Suspend.request`** primitive will nominally be used for this purpose. It shall then use the CFDP **`Put.request`** primitive to request delivery of a file delivery unit to the entity which sent the original Suspend Request message. The metadata of this file delivery unit shall contain one each of the Reserved CFDP Messages of message types Remote Suspend Response (defined in 6.5.4.2) and Originating Transaction ID.

6.5.4.1.2 It is possible for there to be multiple concurrent motivations for suspending a transaction; for example, a CFDP entity might receive a Remote Suspend Request for a transaction which is already suspended due to a locally detected fault condition. The CFDP user application is responsible for managing a queue of such suspension orders in such a way that a suspended transaction remains suspended until all such orders have been explicitly reversed by corresponding resume orders; in particular, a transaction for which a suspension-inducing fault condition was detected must remain suspended until the CFDP user application determines that this fault condition no longer holds, and a transaction for which a remote suspend request was received from a given CFDP entity's user must remain suspended until a remote resume request for that transaction is received from the same user. The manner in which this is accomplished is an implementation matter.

6.5.4.2 Remote Suspend Response

The Remote Suspend Response message shall be constructed as indicated in table 6-21.

Table 6-21: Remote Suspend Response Message

Field	Size (bits)	Values	Comments
Suspension indicator	1	'0' — not suspended '1' — suspended	
Transaction status	2	Same values as defined for transaction status in the ACK PDU (see table 5-8)	
Spare	5	All 'zeroes'	
Spare	1	'0'	This field and the following five fields identify the transaction whose suspension was requested; they are taken from the suspend request.
Length of Entity ID	3		
Spare	1	'0'	
Length of Sequence Number	3		
Source Entity ID	Variable		A binary integer.
Transaction Sequence Number	Variable		A binary integer.

6.6.3.2 Remote Resume Request

The Remote Resume Request message shall be constructed as indicated in table 6-23.

Table 6-23: Remote Resume Request Message

Field	Size (bits)	Values	Comments
Reserved for future use	1	set to '0'	
Length of entity ID	3		Number of octets in source entity ID less one; i.e., '0' means that source entity ID is one octet.
Reserved for future use	1	set to '0'	
Length of Transaction sequence number	3		Number of octets in sequence number less one; i.e., '0' means that sequence number is one octet.
Source entity ID	variable		Uniquely identifies the entity that originated the transaction that is to be resumed. A binary integer.
Transaction sequence number	variable		Uniquely identifies the transaction, among all transactions originated by this entity. A binary integer.

6.6.4 RESPONDING TO A REMOTE RESUME REQUEST

6.6.4.1 General

Upon receipt of a Remote Resume Request message, the remote CFDP user shall assure the resumption of the indicated transaction if possible; the CFDP **`Resume.request`** primitive will nominally be used for this purpose. It shall then use the CFDP **`Put.request`** primitive to request delivery of a file delivery unit to the entity which sent the original Resume Request message. The metadata of this file delivery unit shall contain one each of the Reserved CFDP Messages of message types Remote Resume Response (defined in 6.6.4.2) and Originating Transaction ID.

NOTE – See 6.5.4.1.2 for suspension order queue management requirements.

6.6.4.2 Remote Resume Response

The Remote Resume Response message shall be constructed as indicated in table 6-24.

Table 6-24: Remote Resume Response Message

Field	Size (bits)	Values	Comments
Suspension indicator	1	'0' — not suspended '1' — suspended	Because multiple motivations for suspending a transaction may be concurrently valid, the successful processing of a Remote Resume Request may not actually change the suspension status of the affected transaction.
Transaction status	2	Same values as defined for transaction status in the ACK PDU (see table 5-8)	
Spare	5	All 'zeroes'	
Spare	1	'0'	This field and the following five fields identify the transaction whose resumption was requested; they are taken from the resume request.
Length of Entity ID	3		
Spare	1	'0'	
Length of Sequence Number	3		
Source Entity ID	Variable		A binary integer.
Transaction Sequence Number	Variable		A binary integer.

6.7 STORE AND FORWARD OVERLAY OPERATIONS

6.7.1 GENERAL

The CFDP Store and Forward Overlay (SFO) system is an alternative mechanism for transmitting files between users of CFDP entities which may never be in direct communication; this mechanism does not rely on implementation of the Extended Procedures.

6.7.4.1.3 If the computed route contains only the local CFDP entity, then the user application is the final destination of the file. The attempt to initiate SFO transmission shall therefore immediately fail.

6.7.4.1.4 If the computed route contains only the local CFDP entity and that of the final destination user, then the user application shall simply use the CFDP **`Put.request`** primitive to request direct delivery of the file and associated metadata to the final destination. No further SFO functionality shall apply.

6.7.4.1.5 If the computed route contains one or more other CFDP entities (waypoints) in addition to the local CFDP entity and the final destination user's CFDP entity, then the user application shall use the CFDP **`Put.request`** primitive to request delivery of an SFO transmission file delivery unit to the first waypoint in the route. The file transmitted in the SFO transmission FDU shall be the file that is to be delivered to the final destination user, if any. The metadata of the SFO transmission FDU shall ~~be constructed from the metadata associated with that file and shall comprise~~include a single SFO Request message, defined in 6.7.4.2, together with zero or more of the Reserved CFDP Messages defined in 6.7.4.3 through 6.7.4.6; the contents of these messages shall be obtained from the metadata associated with the file. The destination file name~~, transmission mode, segmentation control, fault handler overrides, and flow label that constrain the resulting transaction shall be those specified in the metadata of the SFO transmission FDU.~~ specified in the **`Put.request`** primitive shall be formed as follows:

SFOsourceEntityID_SFOrequestLabel.sfo

where *SFOsourceEntityID* is the ASCII character representation of the source entity ID noted in the SFO Request message, with all leading zeroes omitted, and *SFOrequestLabel* is as noted in the SFO Request message.

NOTES

1 Path name is omitted from this file name, to avoid incompatibility with path naming rules imposed by the filestores at waypoints.

2 A CFDP user application that implements SFO should reserve space in the local filestore for temporary storage of files that are to be forwarded rather than delivered locally. The action taken with regard to any file received for SFO forwarding that, if retained, would cause this space allocation to be exceeded is implementation-specific.

The transmission mode, segmentation control, fault handler overrides, and flow label that constrain the CFDP transaction resulting from this **`Put.request`** shall be selected at the discretion of the requesting user application.

NOTE – At any time after the transaction finishes, the user application may at its option delete its local copy of the transmitted file.

6.7.4.2 SFO Request

The SFO Request message is mandatory and shall be constructed as indicated in table 6-26.

Table 6-26: SFO Request Message

Field	Size (bits)	Values	Comments
Trace control flag	2	0 (no trace requested), 1 (trace toward source only), 2 (trace toward destination only), 3 (trace in both directions)	Controls the transmission of waypoints' reports of relay transaction success.
Transmission mode	1	As defined in the CFDP Recommendation.	Specifies the transmission mode for ~~each transmission to or by a waypoint~~ transmission from the agent to the final destination.
Segmentation control	1	As defined in the CFDP Recommendation.	Specifies file segmentation control for ~~each transmission to or by a waypoint~~ transmission from the agent to the final destination.
Spare	4		
Prior waypoints count	8	Set to zero by original source user, incremented by every waypoint.	~~Set to zero by original source user, incremented by every waypoint.~~Indicates to the receiver of the request the number of times the request has been forwarded, so far, by waypoint entities.
SFO request label	Variable	LV	A transaction identifier constructed in implementation-specific manner by the original source user. May be used for transaction accounting purposes by the original source user. An array of printable ASCII text characters.

Source entity ID	Variable	LV	The ID of the original source user's CFDP entity. A binary integer.
Destination entity ID	Variable	LV	The ID of the final destination user's CFDP entity. A binary integer.
Source file name	Variable	LV	Length is zero if parameter is omitted.
Destination file name	Variable	LV	Length is zero if parameter is omitted.

6.7.4.3 SFO Message to User

One or more SFO Message to User messages may be optionally included and shall be constructed as indicated in table 6-27.

Table 6-27: SFO Message to User Message

Field	Size (bits)	Values	Comments
Message text	Variable	LV	Encapsulates the text of a message to user that is to be delivered to the final destination user.

6.7.4.4 SFO Filestore Request

One or more SFO Filestore Request messages may be optionally included and shall be constructed as indicated in table 6-28.

Table 6-28: SFO Filestore Request Message

Field	Size (bits)	Values	Comments
Length (octets)	8	Number of octets in the request field.	
Request	8 × length	The value of this field is a single CFDP filestore request as defined in the CFDP Recommendation.	Encapsulates a filestore request that is to be executed by the local CFDP entity of the final destination user.

6.7.4.5 SFO Fault Handler Override

One or more SFO Fault Handler Override messages may optionally be included and shall be constructed as indicated in table 6-29.

Table 6-29: SFO Fault Handler Override Message

Field	Size (bits)	Values	Comments
Fault handler override	8	As defined in the CFDP Recommendation.	~~Applies to each transmission to or by a waypoint.~~Applies to transmission from the agent to the final destination.

6.7.4.6 SFO Flow Label

The SFO Flow Label message is optional and shall be constructed as indicated in table 6-30.

Table 6-30: SFO Flow Label Message

Field	Size (bits)	Values	Comments
Flow label	Variable	LV, as defined in the CFDP Recommendation.	~~Applies to each transmission to or by a waypoint.~~Applies to transmission from the agent to the final destination.

6.7.5 RELAYING AN SFO TRANSMISSION FILE DATA UNIT

6.7.5.1 Upon receipt of a **`Transaction-Finished.indication`** indicating successful reception of a file data unit whose metadata included an SFO Request message, the CFDP user shall invoke the SFO Routing procedure (see 6.7.3) to determine a route from its local CFDP entity to that of the final destination user.

6.7.5.2 If the computed route is an empty list, then relaying is impossible. The CFDP user shall use the SFO Reporting procedure (see 6.7.6) to notify the original source user application of the routing failure.

6.7.5.3 If the computed route contains only the local CFDP entity, then the user application is the final destination of the file. No further SFO functionality shall apply.

6.7.5.4 If the computed route contains one or more waypoints in addition to the local CFDP entity and the final destination user's CFDP entity, then a relay transaction is in order.

6.7.5.5 If the Prior Waypoints Count in the SFO Request Message is equal to the maximum possible value for this field (that is, 255), then relaying is disallowed. The CFDP user shall use the SFO Reporting procedure (see 6.7.6) to notify the original source user application that the maximum number of waypoints was exceeded for this request.

6.7.5.6 If the destination file name specified in the file data unit's metadata does not conform to the convention defined in 6.7.4.1.5 above, then relaying is disallowed. The CFDP user shall use the SFO Reporting procedure (see 6.7.6) to notify the original source user application that the intermediate destination file name was invalid.

6.7.5.7 Otherwise, the user application shall use the CFDP **`Put.request`** primitive to request delivery of the received file (if any) and all received SFO Request, SFO Flow Label, SFO Fault Handler Override, SFO Message to User, and SFO Filestore Request messages to the first waypoint in the route. The destination file name~~,~~specified in the **`Put.request`** primitive shall be the destination file name specified in the received file data unit's metadata. The transmission mode, segmentation control, fault handler overrides, and flow label that constrain the resulting transaction shall be ~~those specified in the metadata of the received FDU~~selected at the discretion of the requesting user application.

6.7.5.8 If the relay transaction finishes in any condition other than 'No error', then the CFDP user shall use the SFO Reporting procedure (see 6.7.6) to notify the original source and final destination user applications of the relay transaction failure.

NOTE – This notification is unconditional, regardless of the value of the trace control flag field in the received SFO Request message.

6.7.5.9 If the relay transaction finishes in 'No error' condition, provided the Trace Control flag in the SFO Request Message is non-zero, the user application shall use the SFO Reporting procedure (see 6.7.6) to report to the original source and/or final destination user applications (as indicated by the value of the Trace Control flag) on the success of the relay transaction.

NOTE – At any time after the relay transaction finishes, the user application may at its option delete its local copy of the transmitted file.

6.7.5.10 If the computed route contains only the local CFDP entity and that of the final destination user, then the user application—now serving as the agent of the SFO transmission

operation—shall initiate an *SFO final delivery transaction*. To do so, it shall use the CFDP **`Put.request`** primitive to request delivery of a file delivery unit with the following parameters:

- destination entity ID as specified in the SFO Request message;
- source file name as specified in the SFO Request message (if any);
- destination file name as specified in the SFO Request message (if any);
- segmentation control as specified in the SFO Request message;
- transmission mode as specified in the SFO Request message;
- flow label as specified in the SFO Flow Label message that was received in the same transaction's **`Metadata-recv.indication`** (if any);
- one fault handler override for each SFO Fault Handler Override message that was received in the same transaction's **`Metadata-recv.indication`**;
- one Message to User for each SFO Message to User message that was received in the same transaction's **`Metadata-recv.indication`**;
- one filestore request for each SFO Filestore Request message that was received in the same transaction's **`Metadata-recv.indication`**;
- the entire SFO Request message, as an additional Message to User. NOTE that this Message to User is, again, a Reserved CFDP Message as defined in 6.7.4.2.

When the resulting transaction finishes, the agent user application shall use the SFO Reporting procedure (see 6.7.6) to report to the original source user application on the success or failure of the final file delivery transaction.

NOTE – At any time after the transaction finishes, the user application may at its option delete its local copy of the transmitted file.

6.7.6 SFO REPORTING

6.7.6.1 General

6.7.6.1.1 When required to report on the results of an SFO file transmission, the user application shall invoke the SFO Routing procedure (see 6.7.3) to determine routes from its local CFDP entity to those of the original source user and/or final destination user as applicable.

6.7.6.1.2 If the computed route to the source is an empty list, then reporting to the source is impossible. No further SFO functionality shall apply.

6.7.6.1.3 If the computed route to the source contains only the local CFDP entity, then the user application is the original source of the file. Therefore reporting to the source is unnecessary, so no further SFO functionality shall apply.

6.7.6.2 SFO Report

The SFO Report message is mandatory in every SFO transmission report and shall be constructed as indicated in table 6-31. If the SFO transmission report is the result of completion of an SFO final delivery transaction, then the message's condition code, delivery code, and file status shall be obtained from the **`Transaction-Finished.indication`** that signaled that completion; otherwise the values of all three fields shall be undefined.

Table 6-31: SFO Report Message

Field	Size (bits)	Values	Comment
SFO request label	Variable	LV	Copied from the request label in the SFO Request message for this SFO transmission.
Source entity ID	Variable	LV	The ID of the original source user's CFDP entity for this SFO transmission. A binary integer.
Destination entity ID	Variable	LV	The ID of the final destination user's CFDP entity for this SFO transmission. A binary integer.
Reporting entity ID	Variable	LV	The ID of the CFDP entity of the user which originally issued this SFO transmission report. A binary integer.
Prior waypoints count	8	Set to zero by the reporting user application, incremented by every waypoint.	Indicates to the receiver of the report the number of times the report has been forwarded, so far, by waypoint entities.
Prior waypoints count of SFO request	8	Copied from the prior waypoints count in the SFO Request message for this SFO transmission.	~~Copied from the prior waypoints count in the SFO Request message for this SFO transmission.~~
Report code	8	1, 2, 3, 4, 5, ~~or~~ 6, or 7	1: final file delivery transaction completed. 2: relay transaction failed. 3: routing failed, relay was not possible. 4: relay transaction succeeded. 5: final file delivery transaction failed. 6: maximum number of waypoints exceeded. 7: invalid intermediate destination file name, relay was not possible.
Condition code	4	See the CFDP Recommendation.	

6.7.7.3 If the computed route contains only the local CFDP entity, then the user application is the original source of the file. No further SFO functionality shall apply.

6.7.7.4 If the computed route contains more than one entity, then ~~the user application shall use the CFDP **`Put.request`** primitive to request delivery of the received SFO Report and SFO Filestore Response message(s) to the second entity in the computed route (the first one following the local CFDP entity)~~a relay transaction is in order.

6.7.7.5 If the Prior Waypoints Count in the SFO Report Message is equal to the maximum possible value for this field (that is, 255), then relaying is disallowed. No further SFO functionality shall apply.

6.7.7.6 Otherwise, the user application shall use the CFDP **`Put.request`** primitive to request delivery of the received SFO Report and any received SFO Filestore Response message(s) to the second entity in the computed route (the first one following the local CFDP entity). When the resulting transaction finishes, or if for any reason the **`Put.request`** cannot be honored, the user application shall take no further action with regard to this SFO transmission.